

An Extended Savings Algorithm for UAS-based Delivery Systems

Younghoon Choi*, Youngjun Choi[†], Simon Briceno[‡], and Dimitri Mavris[§]
Georgia Institute of Technology, Atlanta, Georgia, 30332

This paper presents an extended savings algorithm for a package delivery system using unmanned aircraft systems (UAS). The savings algorithm as a heuristic method solves a vehicle routing problem (VRP) that is commonly formulated by an operational plan for each vehicle. In general, package delivery systems need to establish an operational plan based on demand and preferred time to be visited for each customer. In UAS-based delivery systems, however, capacity and traveling time constraints must be additionally considered to create their operational schedules because of limited payload capacity and short endurance of unmanned aerial vehicles (UAVs). Because of these limitations, UAVs should be reused during operation hours to reduce acquisition costs. Thus, a recharging strategy should be included in the operational planning process. However, conventional savings algorithms cannot capture those properties at once because they have mainly focused on delivery systems with conventional vehicles such as trucks and passenger/cargo aircraft that have different vehicle features and operational characteristics, such as the endurance/speed of a vehicle and recharging strategy. To overcome the limitations of the conventional approaches, this paper proposes the extended savings algorithm, which can concurrently reflect the characteristics of both delivery systems and UAVs. To demonstrate the proposed extended savings algorithm this paper preforms numerical simulations with two representative scenarios in Annapolis, MD and Macon, GA.

I. Introduction

RECENTLY, various research related to UAS have been actively studied because of its benefits such as high agile platform, easy to implement, and low acquisition/operating cost. Because of these benefits, diverse UAS-based applications have received attention. As one of UAS-based applications, the last-mile package delivery using UAVs has been researched because of several advantages. First, a UAV is capable of delivering packages faster than conventional vehicles for package delivery such as trucks or vans since it does not suffer from a complicated road system and the uncertainty of the road system like a traffic jam. Second, a UAS-based delivery system can reduce acquisition and operating cost as compared with conventional vehicles. Because of these benefits of the UAS-based delivery system, many package delivery companies such as Amazon, DHL, UPS, and FedEx have invested on the research associated with the last-mile package delivery using UAVs.

This paper addresses the UAS-based last-mile delivery problem. The problem is that all packages in a single depot should be delivered to customers at their preferred time to be visited. In general, this problem is called the vehicle routing problem with time windows (VRPTW). The result of solving this problem determines the operational schedules of each vehicle. To solve the VRPTW, there are two typical approaches; an exact method and heuristic method [1, 2]. The exact method finds an optimal solution, while it requires high computational resource as the problem size increases because the VRPTW is a NP-hard problem. To overcome this computational issue, the heuristic method can be applied to solve the VRPTW. Although the result of the heuristic method cannot guarantee the optimal solution, its result can be a reasonably good solution within reasonable computational time.

Most of the delivery research have focused on solving the problem operated by the conventional vehicles. However, the UAS-based delivery problem additionally considers two key challenges. The first challenge is that the UAS operation raises an endurance issue because of low battery energy density [3]. Therefore, the problem how to impose the property of UAV's short endurance during an operational planning process is one key challenge in the delivery research. One way is tracing battery power level for each UAV and determines its returning time depending on the battery level [4].

*Graduate Research Assistant, School of Aerospace Engineering, 270 Ferst Dr, Mail Stop 0150, AIAA Student Member

[†]Research Engineer II, School of Aerospace Engineering, 270 Ferst Dr, Mail Stop 0150, AIAA Member

[‡]Senior Research Engineer, School of Aerospace Engineering, 270 Ferst Dr, Mail Stop 0150, AIAA Senior Member

[§]Regents Professor, School of Aerospace Engineering, 270 Ferst Dr, Mail Stop 0150, AIAA Senior Member

However, this approach needs the mathematical model of a battery that precisely characterizes the power consumption profile. Alternative approach is checking flight time for each UAV directly, and then it defines the returning time based on a predefined endurance as an upper bound. The second challenge is how to reflect a recharging process for the reuse of UAVs. Two main issues when modeling a recharging process are places and methods to recharge. As a place to recharge, a recharging station can be considered [5, 6]. In this approach, each UAV can visit a recharging station as conventional vehicles visit a gas station. A depot, also, can be considered as a place to recharge [4, 7, 8]. In this approach, it is assumed that because the UAS-based delivery system serves the last-mile delivery, even if the UAV's batteries can be recharged only in the depot, the UAVs can deliver packages to the customers in its operational area. On the other hand, the recharging can be conducted by directly recharging batteries combined to UAVs or by swapping batteries. The former should wait until finishing to recharge, while the latter can operate with short recharging time, but it could require a large supply of batteries.

To address these challenges, we propose an extended savings algorithm for UAS-based delivery systems. The savings algorithm proposed by Clarke and Wright [9] is a heuristic method for VRPs (See [10] to compare a variety of variants of conventional savings algorithms). Mirshekarian and Süer [11] present a savings algorithm for a VRPTW. However, these existing algorithms cannot address the challenges that come from UAV's inherent properties since their approach is based on the conventional delivery method. To solve this limitation of savings algorithms, the proposed extended savings algorithm includes endurance constraints and a single-depot recharging strategy.

The main contribution of this paper introduces a new extended savings algorithm for UAS-based delivery systems. The proposed algorithm contains UAV's physical characteristics such as small payload capacity and short endurance, which force UAVs to return to the depot due to their limitation. The method additionally considers their reusability during operating hours, which could reduce the acquisition cost of the delivery systems through decreasing the required number of UAVs.

This paper is organized as follows: Section II introduces UAS-based delivery systems. Section III discusses the extended savings algorithm which can handle the characteristics of UAS-based delivery systems. Section IV conducts numerical simulations with both Annapolis, MD, and Macon, GA, use-cases. This paper ends with conclusions in Section V.

II. Delivery Systems based on UAS

This section presents the key characteristics of UAS-based delivery systems: some come from UAV's inherent natures, and the others originate from operational concepts of the delivery system. For the UAS-based package delivery, an operational plan must entail the UAS characteristics. The primary UAS features associated with a package delivery operation consist of five types; small payload capacity, short endurance, recharging batteries, independence of conventional road systems, and regulations of the Federal Aviation Administration (FAA).

First, a UAV has small payload capacity. Even a UAV could be allowed to convey just a single package at once. This nature makes two operational concepts; single-package delivery operations and multi-package delivery operations. If UAVs cannot deliver multiple packages enough to offset increased expense which is the cost difference for acquisitions and operations between UAVs for single-package delivery and that for multi-package delivery, single-package delivery operations have a benefit in cost.

Second, UAV's endurance is relatively short in comparison with operating hours of its delivery system. In the conventional savings algorithms, each vehicle operates once during operating hours. For instance, let us say that there is a UAS-based delivery system which operates 60-minute-endurance UAVs. If its operating hours is 8 hours, each UAVs stays for 7 hours at least from an operational plan created by the conventional approaches. This operational plan is significantly inefficient in operations. Thus, reuse strategies should be included in operational planning process. In VRPs, a single operation of vehicle is called a trip which starts from a depot and ends at a depot. However, if reuse of vehicles is considered, each vehicle could have more than one trip. To mention all routes allocated to a vehicle, let us call it a journey which is a combination of trips.

Third, UAVs operate by electrical power with rechargeable batteries. When constructing journeys, recharging time also should be considered to find available vehicles at specific time. In the recharging location issue, there are two approaches: one is setting up recharging stations in operational area. The other is allowing UAVs to recharge their batteries only at a depot with reloading packages. In the recharging method issue, UAVs could remain at its recharging location up to finishing recharging batteries, or they could stay during swapping batteries. If swapping batteries is considered, the used batteries could be charged at the recharging location, even though UAVs leave the location after swapping batteries.

Fourth, UAVs can move without suffering from complicated road systems. That is, UAVs are free from topologies of roads, traffic signals, and traffic jams. This property is a key advantage of UAS-based delivery systems. Based on this nature, UAS-based delivery companies are capable of creating more profits by delivering packages faster or delivering packages at specific time. In an operational point of view, delivering packages at specific time can be modeled as the VRPTW. In VRPTWs, the preferred time to be visited for each customers can be described as time windows.

Finally, UAVs should operate under regulations. In the United States, the regulations are being developing by the FAA. Under regulations, the visual line-of-sight (VLOS) operation is required, which forces each vehicle to be watched by a single pilot or observer. Because this is a substantial limitation on UAS-based delivery systems, we assume that the beyond visual line-of-sight (BVLOS) operation is allowed. Additionally, we assume that UAV's technologies are developed enough to solve other operational issues to focus on operational concepts of UAS-based delivery systems.

In this paper, a single-depot UAS-based delivery systems with homogeneous UAVs is considered with the assumptions that; first, each UAVs has relatively small payload capacity and endurance to prevent that a UAVs visits too many customers. Second, every UAV is allowed to swap its batteries and to reload packages at the depot only. Swapping batteries makes the problem simpler because it can be modeled as an activity requiring fixed time. Third, UAVs can move straightly between the depot and a delivery location, or between delivery locations. This implies each UAV operates with simple mission profiles which consists of takeoff, cruise, and landing. Finally, all UAVs operate fully automatically under BVLOS operation conditions. In Section III, these operational concepts are implemented as the extended savings algorithm to build an operational plan for the UAS-based delivery system.

III. Extended Savings Algorithm

This section presents an extended savings algorithm for UAS-based delivery systems that can reflect the UAV's natures described in Section II. The proposed algorithm deals with the multi-trip vehicle routing problem with time windows (MTVRPTW), which is a extended variant of the VRPTW allowing each UAV to have multiple trips or a journey. Based on the graph theory, a MTVRPTW can be described by a graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ and a set of vehicles \mathcal{V} , where \mathcal{N} is a set of all nodes in \mathcal{G} , and \mathcal{A} is a set of all arcs in \mathcal{G} . A set of nodes, $\mathcal{N} = \{0, 1, \dots, n, n+1\}$, consists of a set of two depots, $depot = \{0, n+1\}$, and a set of n customers, $\mathcal{C} = \{1, 2, \dots, n\}$. Note that the two depots indicate the same depot, but $depot_0$ is used when a vehicle leaves the depot, while $depot_{n+1}$ is utilized when a vehicle arrives at the depot. Each customer C_i has preferred time to be visited that is called time window $\mathcal{TW}_i = [a_i, b_i]$, which a_i is ready time, and b_i is due time. Also, each customer has service time s_i , which is the total time of landing, delivering, and takeoff. Each arc \mathcal{A}_{ij} has a distance between \mathcal{N}_i and \mathcal{N}_j , \mathcal{D}_{ij} which is symmetric. Finally, each vehicle \mathcal{V}_i can take a journey \mathcal{J}_i which is a combination of trips in \mathcal{T} . With these notions, this section discusses key rules of the proposed method and present the details of the algorithm structure.

A. Key Rules of the Extended Saving Algorithm

To convert the key characteristics of delivery system using UAVs mentioned in Section II into a savings algorithm, the rules that can be applied to the algorithm need to be created. First, the properties of small payload capacity, short endurance and recharging of UAVs can be modeled mathematical expressions. Second, the straight path of UAVs can be reflected by the straight flight distance among nodes. Last, by the assumption of BVLOS operations and fully advanced technologies of UAVs, this problem can be handled from the point of view of high-level operations of UAVs-based delivery system. Additionally, in the proposed method, one more characteristic is infused, which comes from a limitation of the VRPTW model not a property of UAS-based delivery system. In the VRPTW models, if a vehicle arrives at a delivery location before the designated time called ready time, then the vehicle should wait until ready time at the place. If this concept applies to UAVs, they could be wait a long time on the ground or even in the air by hovering. Because the long time hovering may occur some safety issue, a rule to prevent a long-time hovering is combined.

The proposed method, thus, is based on five key rules: saving rule, time window rule, maximum hovering rule, maximum endurance rule, and multi-trip rule. Saving rule is the fundamental idea of all savings algorithms, which provides a priority order of a pair of nodes to be combined. Time window rule describes how to select a customer to be visited earlier using preferred time to be visited for each customer. Maximum hovering rule prohibits long waiting at a delivery location. Endurance rules are directly related with UAV's properties, which guarantees that each UAV returns at the depot within UAV's maximum endurance. Lastly, multi-trip rule needs to be to build a journey that is a combination of trips assigned to a UAV.

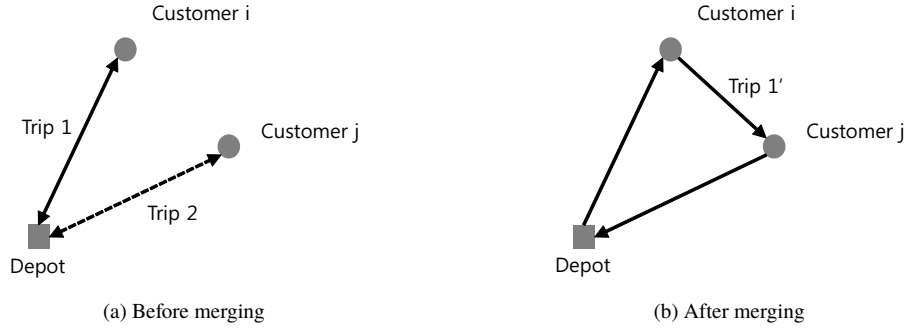


Fig. 1 An example of merging two trips

1. Saving Rule

The saving implies how much distance can be saved when two trips merge into one trip. For instance, there are two trips that each trip visits a customer respectively as shown in Fig. 1a. Then, total distance vehicles should move is calculated by

$$\text{Total Distance}_{[\tau_1, \tau_2]} = \mathcal{D}_{0i} + \mathcal{D}_{i(n+1)} + \mathcal{D}_{0j} + \mathcal{D}_{j(n+1)} = 2\mathcal{D}_{0i} + 2\mathcal{D}_{0j}. \quad (1)$$

If the two trips merge to form a new trip like Fig. 1b, the total distance is updated by

$$\text{Total Distance}_{[\tau_{1'}]} = \mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{j(n+1)} = \mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{0j}. \quad (2)$$

Then, the distance that can reduce by merging the two trips, which is called saving, is computed by

$$\begin{aligned} \mathcal{SV}_{ij} &= \text{Total Distance}_{[\tau_1, \tau_2]} - \text{Total Distance}_{[\tau_{1'}]} \\ &= (2\mathcal{D}_{0i} + 2\mathcal{D}_{0j}) - (\mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{0j}) \\ &= \mathcal{D}_{0i} + \mathcal{D}_{0j} - \mathcal{D}_{ij}. \end{aligned} \quad (3)$$

In the savings algorithm, a pair of trips with high saving value has a priority to merge. Note that when merging trips, at least one of them should be trivial trip which is a trip to visit only one customer.

2. Time Window Rule

In MTVRPTWs, each customer has preferred time to be visited, which is captured by a time window. The time windows determine the order to visit customers. For example, assume that there are a customer i and customer j who should be visited. Because each customer has own time window, three scenarios could occur: in the first scenario, the time window of the customer i start earlier, and even if a UAV moves from the customer i to the customer j after serving the customer i, it can arrive at the customer j within its time window as shown in Fig. 2a. In this case, the UAV can serve the two customers on a trip by visiting the customer i earlier. The second scenario is the inverse situation of the first scenario as illustrated in Fig. 2b. In the same manner, a UAV can serve the two customers with the reverse order to visit the customers. In the last scenario, if the customer j has a time window which it is impossible to be visited after visiting the customer i, and the customer i has a time window which it is also impracticable to be visited after visiting the customer j, then it is unrealizable to allocate two visits for a UAV. Thus, in that case, two trip should remain as shown in Fig. 2c.

3. Maximum Hovering Rule

Although a UAV can visit two customers on a trip based on their time windows, it is possible that the UAV should stay in the air until ready time of the second customer. In this case, as hovering time increases, a safety issue could be emerged with anxieties of neighbors because of noise and privacy issues. In addition, by restricting UAVs to have long hovering, they are forced to return to the depot more frequently to complete the entire mission, but this could increase the opportunity of merging trips to build a journey. Thus, we assume that if a UAV has longer hovering time to visit a customer than the prefixed maximum hovering time, it cannot visit the customer as illustrated in Fig. 3.

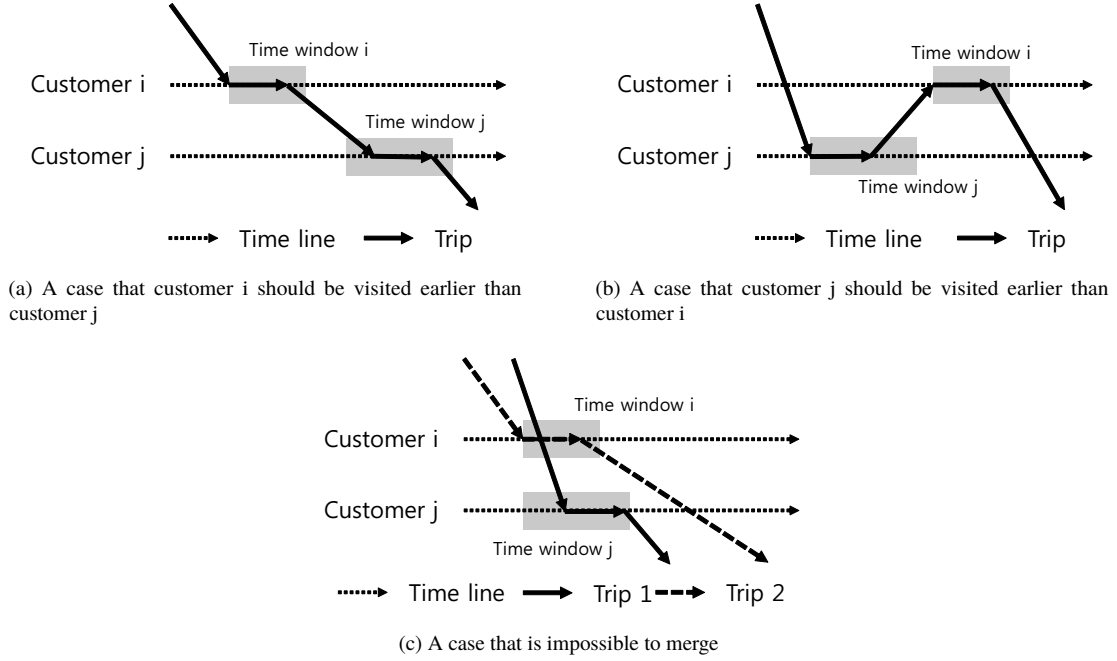


Fig. 2 The effect of time window on the order of visiting customers

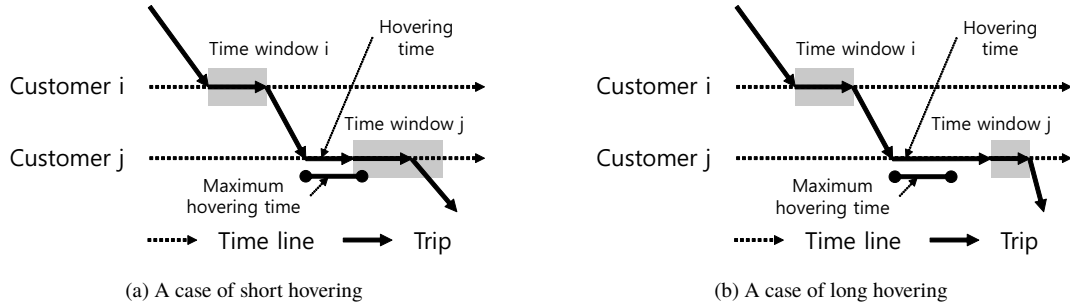


Fig. 3 Maximum hovering rule

4. Maximum Endurance Rule

In conventional savings algorithms, only payload capacity has been used to check whether or not a vehicle can visit a next customer. However, because a UAV has much shorter endurance, it should often visit a location where its battery can be swapped. If the flight time of a trip becomes longer than maximum endurance of the UAV by visiting more customers, the trip should not be extended as an infeasible case as shown in Fig. 4.

5. Multi-Trip Rule

Allowing UAVs to have multiple trips is a key point to significantly reduce the acquisition cost of UAS-based delivery systems. By reusing UAVs, the minimum required number of UAVs decreases, which yields the maximum available number of UAVs concurrently. In order to merge two trips to create a journey, at least there is time enough to reload packages and to swap UAV's batteries as illustrated in Fig. 5.

B. The Algorithm Structure of the Extended Savings Algorithm

The extended savings algorithm consists of three steps; computing savings, creating trips, and building journeys. Each step sequentially works from computing savings to building journeys. The first step finds a set of feasible arcs to

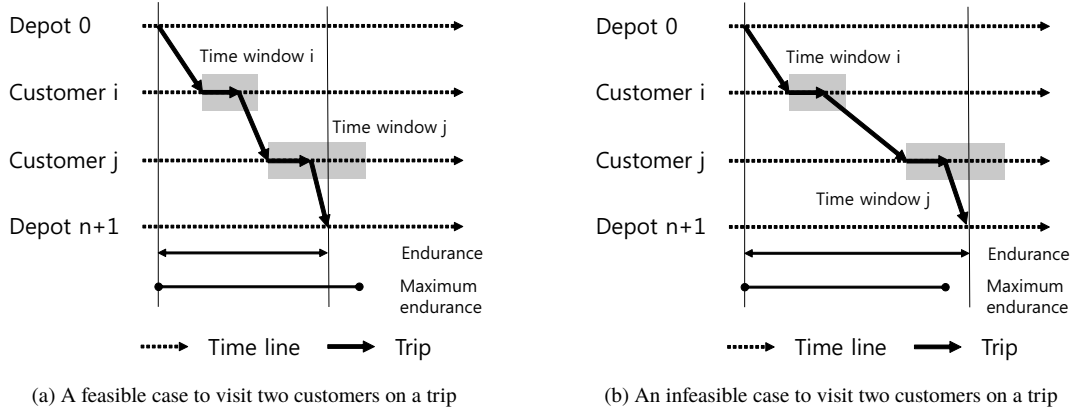


Fig. 4 Maximum endurance rule

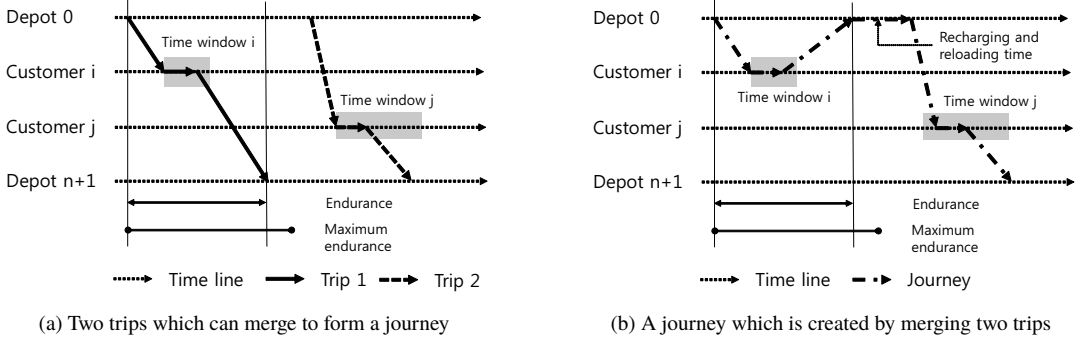


Fig. 5 Multi-trip rule

be capable of merging with their own savings. For feasibility checks, time window rule and maximum hovering rule are applied. If an arc is infeasible, the two customers on the arc cannot be visited sequentially on a trip. For feasible arcs, saving is computed by saving rule, Eq. (3). We note that before looking over each arc, if there is the sorted list of customers based on time windows, computational power can be saved.

In the second step, trivial trips need to be made first, which visit just one customer. The set of trivial trips is a solution of the single-package delivery operation. Thus, if the single-package delivery operation is considered, this step can be skipped. After creating trivial trips, the list of arcs based on saving values is sorted, which implies the priority to merge. Then, during looking over each arc sorted by saving values, time window rule, endurance rule, payload capability check are utilized to find a pair of trip that can merge. Note that when merging two trips, at least one of them should be a trivial trip. In the conventional savings algorithms, the algorithm ends by returning trips.

In the last step, journeys which are allocated to each UAV are built from the trips created in the second step. Each journey has at least one trip, because it is a combination of trips. In this step, sorted trip list is utilized as a priority to merge. Finally, during looking over each trip, journeys are built by journey rule. Note that each journey is assigned to each UAV. The procedure of the proposed algorithm is described as pseudo codes in Algorithm 1.

C. Merge Strategies: Sequential vs. Parallel

When merging two routes into one route, two approaches can be used; sequential and parallel merge strategies. First, the sequential merge strategy works with a single base route which has the largest saving value. The approach extends the base route as much as possible, then it considers a next base route. Second, the parallel merge strategy unites two routes along the order of a sorted list of saving values without considering a base route. The literature written by Laporte et al. reports that the parallel merge strategy generally creates better routes than the corresponding sequential merge strategy [12].

Algorithm 1 Pseudo code of the proposed extended savings algorithm

Input: $\mathcal{G}(\mathcal{N}, \mathcal{A})$ and \mathcal{V} **Output:** \mathcal{J}

< STEP 1: computing savings using saving, time window, and maximum hovering rules >

Initialization

Sort nodes based on the order of time windows

for i in \mathcal{C} **do** **for** j in \mathcal{C} **do** **if** the order to be visited and max hovering time constraint are feasible **then** $\mathcal{SV}_{ij} = \mathcal{D}_{0i} + \mathcal{D}_{0j} - \mathcal{D}_{ij}$ **end if** **end for****end for**

< STEP 2: creating trips using time window and maximum endurance rules >

Create \mathcal{T} with trivial trips for each customerSort \mathcal{A} by descending order based on \mathcal{SV} **for** (i, j) in \mathcal{A} **do** Find trips that visit C_i and C_j respectively **if** at least either C_i or C_j is in a trivial trip **then**

Estimate time schedule and payload capacity to be used after merging two trips

if time schedule, payload capacity, endurance constraints are feasible **then**

Merge two trips

end if **end if****end for**

< STEP 3: building journeys using time window and multi-trip rules >

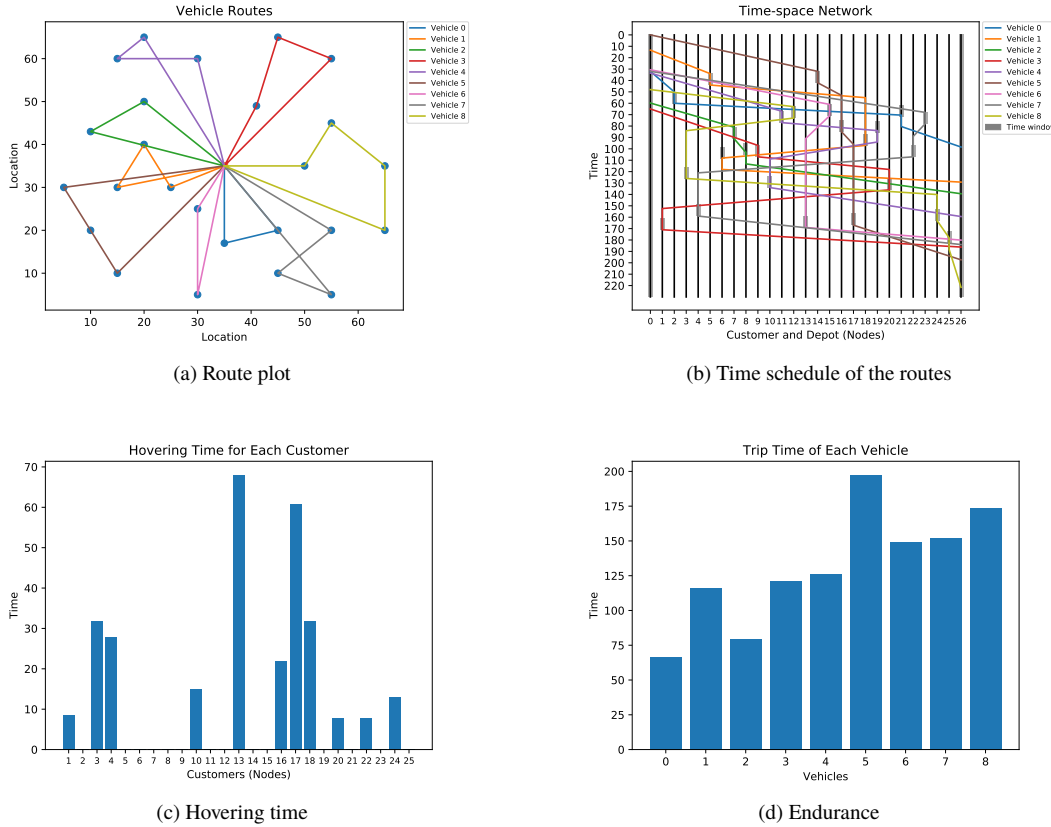
 \mathcal{J} = empty setSort \mathcal{T} based on the order of departure time**for** \mathcal{T}_i in \mathcal{T} **do** **for** \mathcal{J}_i in \mathcal{J} **do** **if** \mathcal{T}_i can be merged into \mathcal{J}_i **then** Update \mathcal{J}_i as \mathcal{J}_i with \mathcal{T}_i **end if** **end for** **if** there is no journey which \mathcal{T}_i can be merged into **then** Add a journey as \mathcal{T}_i into \mathcal{J} **end if****end for****return** \mathcal{J}

For instance, let us consider four trivial routes such that $[0, 1, 5]$, $[0, 2, 5]$, $[0, 3, 5]$, and $[0, 4, 5]$, where 0 and 5 indicate a depot, and the others are delivery locations. It is also assumed that a sorted list of saving values is given such that $(1, 2): 10$, $(3, 4): 8$, $(2, 4): 7$, which means when two nodes in parenthesis are merged, the saving value after colon can be obtained. In the sequential merge approach, $[0, 1, 5]$ and $[0, 2, 5]$ routes can be combined into $[0, 1, 2, 5]$ called a base route. Then, the pair of node 3 and 4 in the list is skipped because it cannot be used to extend the base route. Next, the route $[0, 4, 5]$ is merged into the base route. As the result, the algorithm returns $[0, 1, 2, 4, 5]$ and $[0, 3, 5]$ routes as a solution with the total saving value, 17.

On the other hand, in the parallel merge approach, after combining $[0, 1, 5]$ and $[0, 2, 5]$ routes into $[0, 1, 2, 5]$ route,

Table 1 Summary of numerical simulation of the canonical problem

Method	# of nodes	Merge strategy	# of trips	# of UAVs	Max hovering time	Max endurance
Conventional	25	Parallel	9	9	68	197
Extended	25	Sequential	15	11	4	133
		Parallel	14	10	4	145

**Fig. 6 Results of a conventional savings algorithm for R101.25**

$[0, 3, 4, 5]$ route is created by merging $[0, 3, 5]$ and $[0, 4, 5]$ routes rather than $[0, 1, 2, 4, 5]$ route in the sequential merge approach. This behavior is expected because the saving value from merging $[0, 3, 5]$ and $[0, 4, 5]$ routes is larger than that from combining $[0, 1, 2, 5]$ and $[0, 4, 5]$ routes. That is, the parallel approach tries to maximize total saving value by combining with multiple routes at once along the sorted list of saving values. Next, the algorithm considers to unite two route visiting node 2 and 4 respectively. However, node 2 and 4 cannot be connected because both are not trivial routes. Thus, the parallel approach returns $[0, 1, 2, 5]$ and $[0, 3, 4, 5]$ routes as a solution with total saving value, 18.

IV. Numerical Simulations

A. Study Case: a Canonical Problem

In this section, computational simulations employing one of Solomon's benchmarking data set for the VRPTW*, which is called R101.25, are conducted to demonstrate the extended savings algorithm. The test scenarios are that: first,

*<http://w.cba.neu.edu/~msolomon/problems.htm> (accessed 5 November 2018)

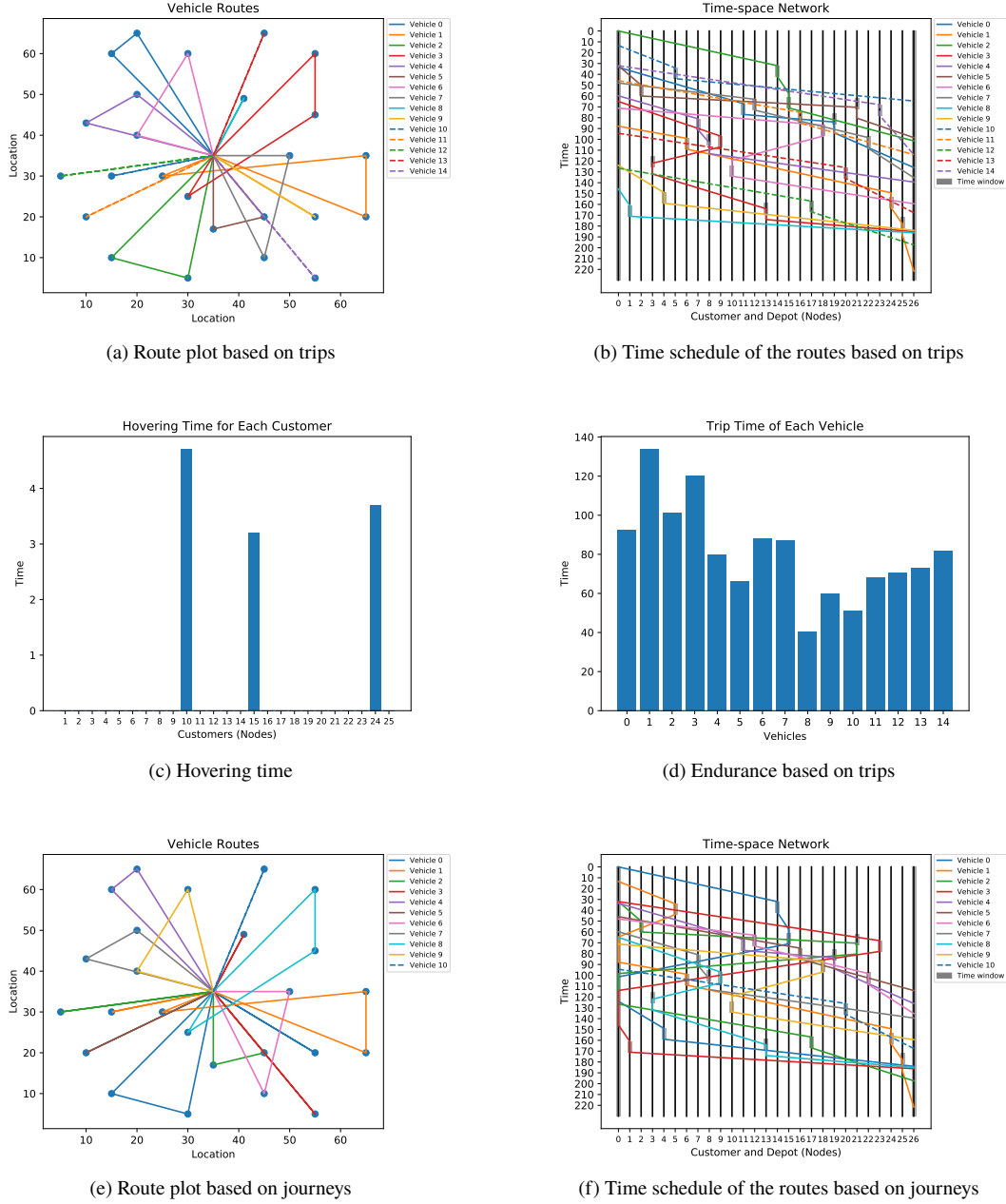


Fig. 7 Results of the extended sequential savings algorithm for R101.25

there are a single depot and 25 customers. Each customers has own time window and package demand with 10 service time. This benchmark data set uses unit time and unit distance. Each vehicle moves unit distance at unit velocity, and it is capable of carrying packages up to maximum payload capacity. For the extended algorithm, let us assume that: first, the preferred maximum hovering time is 5 unit time. Second, maximum endurance of the UAV is 150 unit time. With those conditions, Computational simulations for a conventional savings algorithm and the extended savings algorithm are conducted respectively. All the numerical simulations are executed with Intel® Core™ i7-7700HQ processor and 32 Gb memory. The computational result is summarized in Table 1.

As a baseline, the conventional savings algorithm for the VRPTW is employed, which does not include maximum hovering, maximum endurance, and journey rules. The solution of the VRPTW can be reconstructed as a route plot and

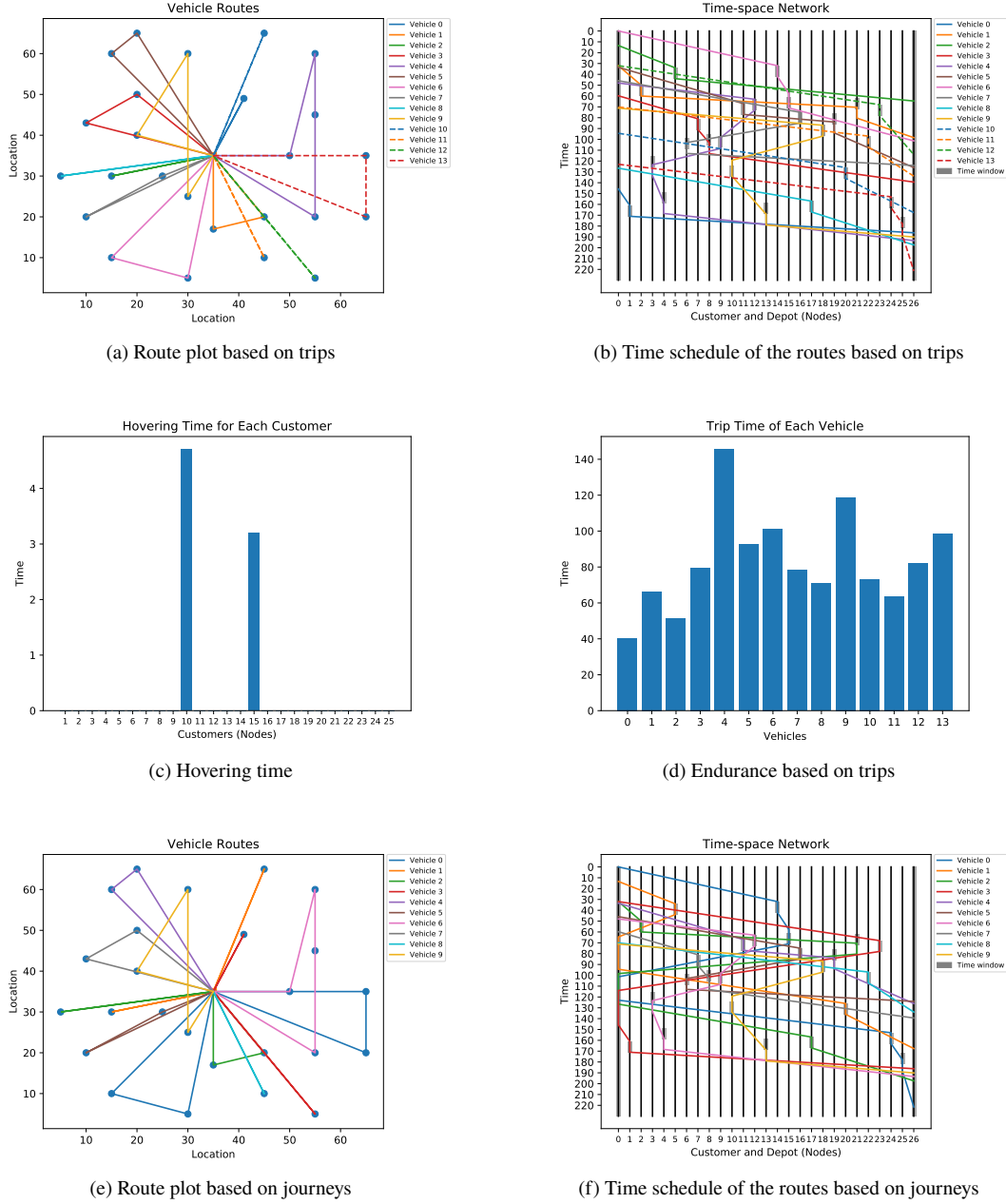


Fig. 8 Results of the extended parallel savings algorithm for R101.25

a time schedule. The results of the conventional savings algorithm are illustrated in Fig. 6. The route plot shows how the vehicles move spatially like Fig. 6a. In this case, to serve 25 customers, 9 vehicles need to be in the UAS-based delivery system. Then, the temporal information of each vehicles can be obtained in the time schedule as shown in Fig. 6b. Note that x-axis of the plot shows just a list of nodes, which implies that it does not provide any spatial information. In conventional savings algorithm, because there is no rules of maximum hovering and maximum endurance, longer hovering than the preferred maximum hovering time occurs as presented in 6c. Moreover, some routes require longer trip time than maximum endurance of the UAV. Note that this result clearly shows that the conventional savings algorithms have a limitation to apply the UAS-based delivery problems.

The simulation results of the extended savings algorithm with sequential merge strategy are described in Fig. 7. In

the spatial and temporal results as shown in 7a and 7b, they require the more number of vehicles than the conventional algorithm. However, as each trip satisfies the constraints of both maximum hovering time rule as shown in Fig. 7c and 7d. This result represents that the proposed algorithm has a capability of dealing with UAS-based delivery problems by directly handling the characteristics of both the UAV and the delivery system. Moreover, by applying journey rule, the required number of UAVs can be reduced as illustrated in Fig. 7e and 7f. This is a key characteristic of multi-trip algorithms, which allows to build an operational plan for entire hours of operation with a minimum of vehicles.

On the other hand, the results from the proposed savings algorithm with the parallel merge strategy are illustrated in Fig. 8. The solution of the parallel extended savings algorithm, also, satisfies maximum hovering, maximum endurance, and journey rules. However, the solution is different from that of the sequential algorithm. This result shows the merge strategy has an impact on creating building routes. In this canonical example, the parallel savings algorithm creates a better solution by requiring the less number of vehicle to complete the given delivery tasks: the parallel merge strategy requires 10 vehicles whereas the sequential strategy needs 11 vehicles.

B. Study Case: Two UAS-based Delivery Systems in Annapolis, MD, and Macon, GA

As UAVs can fly more straightly among nodes, the benefits of UAS-based delivery systems increase. If there are obstacles such as tall buildings at the operational altitude, UAVs should make a detour to avoid the obstacles, which makes routes worse in flight distance and time. From Part 107 of the FAA, small UAVs under 55 lb can operate at or below 400 ft [†]. In order to reduce the effects of urban obstacles such as tall buildings on vehicle's routes, both Annapolis, MD, and Macon, GA, cities are selected for numerical simulations of UAS-based delivery scenarios because an operational altitude without obstacles can be utilized: the tallest building in Annapolis, MD, is about 98 ft with 8 floors [‡], whereas the one in Macon, GA, is about 215 ft with 15 floors [§]. Each city scenario is simulated with both the sequential and parallel extended savings algorithms.

In the Annapolis scenario, there are a depot, which is a USPS distribution center, and 51 delivery locations consisting of gas stations and USPS offices with 161 time windows, which implies that this scenario has 163 nodes. The depot and delivery locations in Annapolis, MD, are shown in Fig. 9a and 9b. The distribution is illustrated with a Probability Density Function (PDF) estimated by the Kernel Density Estimation (KDE) which is a non-parametric approach to estimate a PDF of random variable. In the Macon scenario, a depot, which is a FedEx ship center, and 139 delivery locations which are gas stations are selected with 466 time windows, which consists of 468 nodes. The depot and delivery locations in Macon, GA, are illustrated in Fig. 9c and 9d. In both scenarios, most of delivery locations are within 10 mi from the depot, but delivery locations in Macon, GA, are relatively far away from the depot. Each delivery location has a demand for every time window, which are created by a uniform distribution as described in Fig. 10. The common assumptions for both simulations are summarized below.

- Operational altitude of UAVs is 300 ft.
- UAVs fly straightly among the depot and nodes.
- Delivery systems operate for 8 hours.
- UAVs operate at speed of 50 mph.
- Maximum payload capacity of UAVs is 10 lb.
- Maximum demand of delivery location is 5 lb.
- Maximum endurance of UAVs is 60 mins.
- Maximum hovering time is 5 mins.
- Time for reloading packages and swapping batteries is 15 mins.
- Each delivery location can have at most 5 time windows.
- Each time window is a period of time from 5 to 30 mins.
- Time windows at a delivery location cannot be overlapped.
- UAVs spend 3 mins at each delivery location including landing, delivery, and take-off.

The numerical simulation results are summarized in Table 2. As a mid-size problem, the Annapolis scenario has 161 nodes, whereas the Macon scenario has 466 nodes as a large-size problem. In the Annapolis scenario, the sequential merge strategy requires 11 UAVs while the parallel merge strategy needs 10 UAVs to deliver all packages. In the Macon

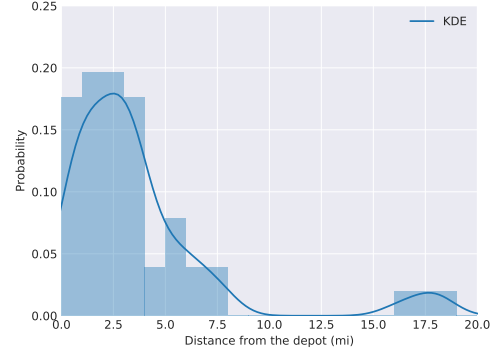
[†]https://www.faa.gov/uas/getting_started/part_107/ (accessed 5 November 2018)

[‡]<https://www.emporis.com/statistics/tallest-buildings/city/101532/annapolis-md-usa> (accessed 5 November 2018)

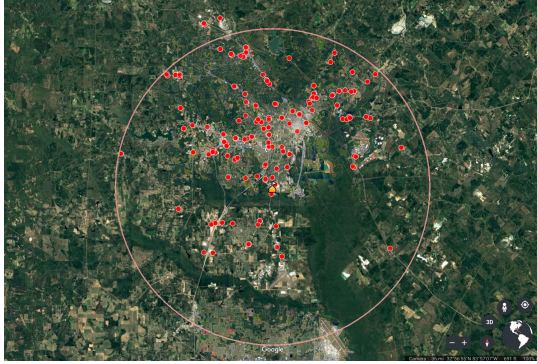
[§]<https://www.emporis.com/statistics/tallest-buildings/city/101707/macon-ga-usa> (accessed 5 November 2018)



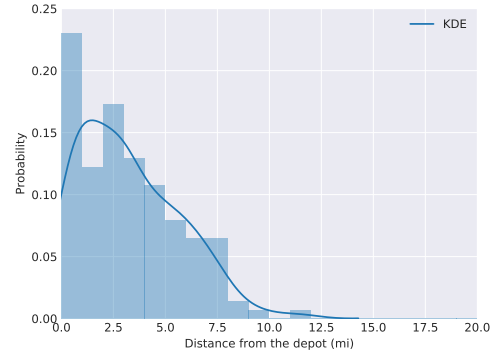
(a) Delivery locations of the Annapolis scenario with a 10 mi radius circle at the depot (Google Earth)



(b) Delivery location distribution of the Annapolis scenario

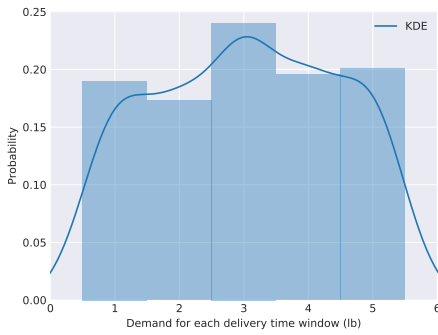


(c) Delivery locations of the Macon scenario with a 10 mi radius circle at the depot (Google Earth)

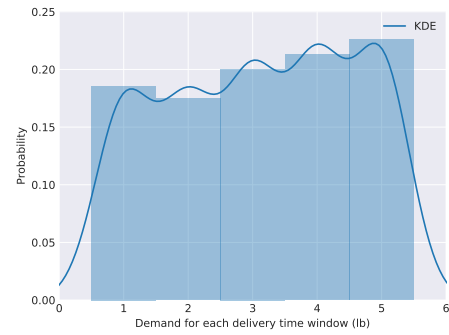


(d) Delivery location distribution of the Macon scenario

Fig. 9 Spatial information of two simulation scenarios



(a) The Annapolis scenario



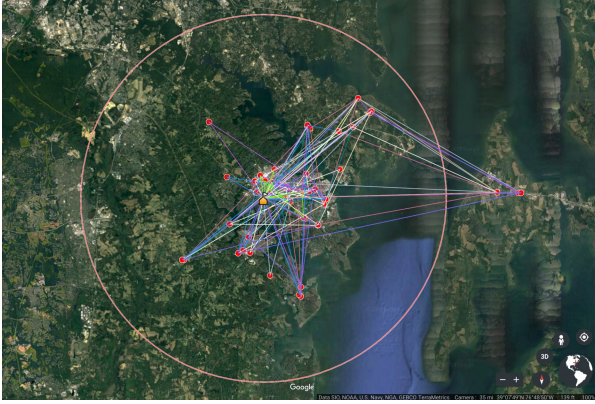
(b) The Macon scenario

Fig. 10 Demand distributions of simulation scenarios

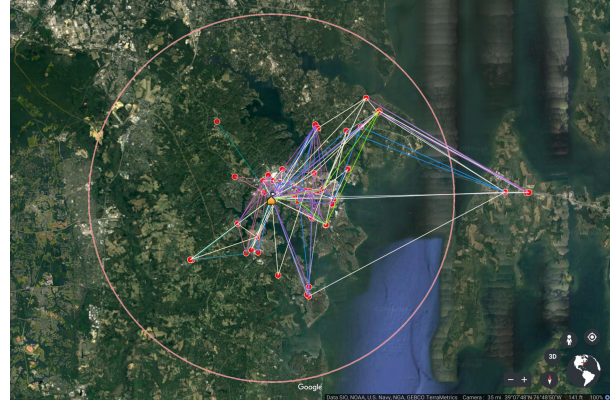
scenario, the solution of the sequential approach requires 25 UAVs while the solution of the parallel approach needs 18 UAVs. Those results show that the solution of the parallel merge strategy dominates the solution of the sequential merge strategy. The journeys of each solution are illustrated in Fig. 11, but many lines for each journey are overlapped, thus

Table 2 Summary of numerical simulation of city scenarios

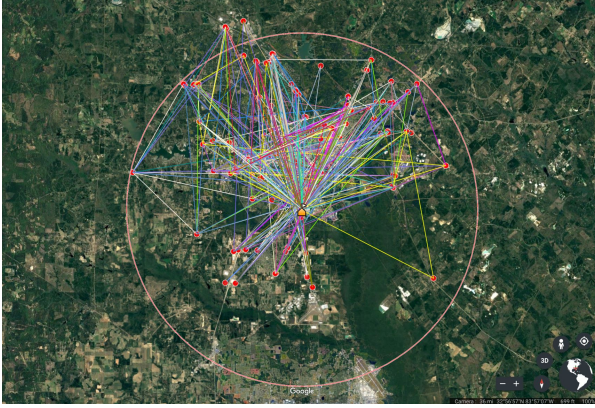
City	# of delivery locations	# of nodes	Merge strategy	# of trips	# of UAVs	Elapsed time (sec)		
						First	Second	Third
Annapolis, MD	51	161	Sequential	68	11	8.630	8.690	9.012
			Parallel	59	10	0.709	0.718	0.706
Macon, GA	139	466	Sequential	168	25	421.489	418.161	417.859
			Parallel	134	18	6.631	6.584	6.480



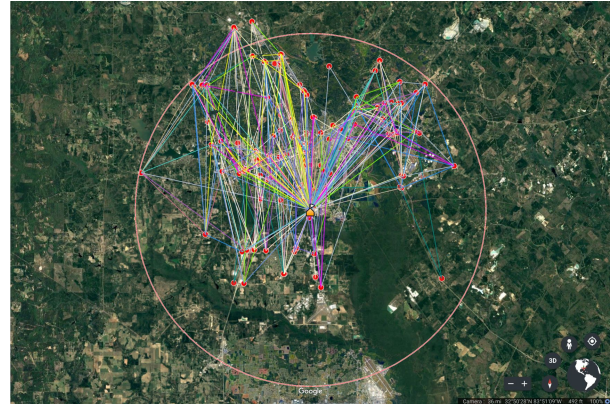
(a) The Annapolis scenario with sequential merge strategy: 11 UAVs



(b) The Annapolis scenario with parallel merge strategy: 10 UAVs



(c) The Macon scenario with sequential merge strategy: 25 UAVs

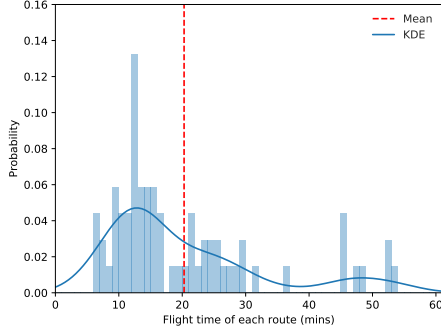


(d) The Macon scenario with parallel merge strategy: 18 UAVs

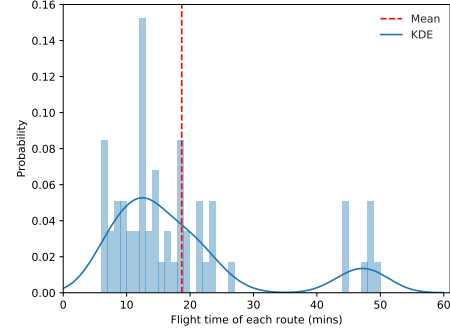
Fig. 11 Solution journeys

the solutions are reconstructed as a distribution of flight time as described in Fig. 12. The shape of PDFs of the trip distribution are similar with those of the node distribution. However, the PDFs of the parallel merge strategy in Fig. 12b and 12d is denser than that of the sequential merge strategy in Fig. 12a and 12c with a lower mean value. This implies that the parallel algorithm creates a solution requiring the total flight hours less than the sequential algorithm.

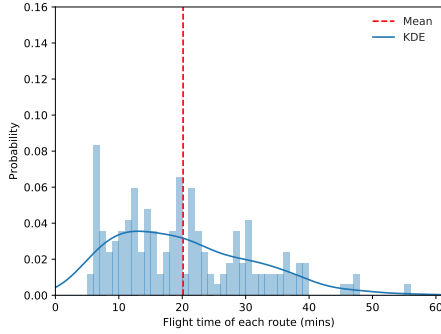
Each simulation is executed three times as shown in Table 2. The parallel approach is faster than the sequential approach as well as it has more potentialities to solve a larger size problem: the elapsed time of the parallel approach increases up to 9.23 times when solving the problem with the Macon scenario based on the Annapolis scenario whereas that of the sequential approach increases up to 47.74 times.



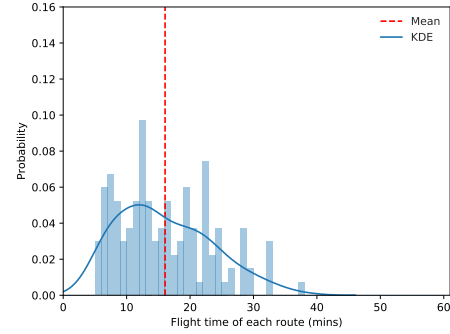
(a) The Annapolis scenario with sequential merge strategy



(b) The Annapolis scenario with parallel merge strategy



(c) The Macon scenario with sequential merge strategy



(d) The Macon scenario with parallel merge strategy

Fig. 12 Trip distribution of solution based on flight time

V. Conclusions

This paper proposed an extended savings algorithm to handle UAS-based package delivery problem. For the operational planning of UAS-based delivery systems, the characteristics of UAVs must be considered as well as that of the logistics to guarantee the feasible operational route and schedule, which precisely takes into account limited payload capacity and short endurance requirement. Existing conventional savings algorithms for the VRPTW can deal with the characteristics of the logistics, but they cannot reflect the properties of UAVs. Therefore, this paper introduces five rules of UAS-based delivery systems: saving rule, time window rule, maximum hovering rule, maximum endurance rule, and journey rule. Based on these rules, this paper formulates an extended savings algorithm for UAS-based delivery systems, which is capable of capturing the features of both vehicles and the logistics to generate an operational plan.

Through the numerical simulations, this paper shows that the proposed extended savings algorithm is more suitable to handle UAS-based delivery problems than the conventional savings algorithms because the proposed algorithm considers key five UAS operational rules: maximum hovering rule, maximum endurance rule, and journey rule. Moreover, this paper compares the solution of the sequential merge strategy with that of the parallel merge strategy. The results present that the parallel merge strategy leads a better solution with less computational time.

One of the potential extensions of this work is to add more practical logistic aspects such as multiple depots or recharging stations for UAS-based delivery systems to cover wider area. Another possible extension is to reformulate energy-based savings algorithm rather than maximum endurance of UAVs because UAS endurance can vary according to flight status. Although the energy-based algorithm requires more complexity in the algorithm, it can produce more accurate solutions for real UAS operations.

References

- [1] Eksioglu, B., vural, A. V., and Reisman, A., “The Vehicle Routing Problem: A Taxonomic Review,” *Computers & Industrial Engineering*, Vol. 57, No. 4, 2009, pp. 1472–1483.
- [2] Laporte, G., “The Vehicle Routing Problem: an Overview of Exact and Approximate Algorithms,” *European Journal of Operational Research*, Vol. 59, No. 3, 1992, pp. 345–358.
- [3] Choi, Y., Choi, Y., Briceno, S., and Mavris, D. N., “Three-dimensional uas trajectory optimization for remote sensing in an irregular terrain environment,” *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 1101–1108.
- [4] Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S., “Vehicle Routing Problems for Drone Delivery,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, 2017, pp. 70–85. doi:10.1109/TSMC.2016.2582745.
- [5] Erdoğan, S., and Miller-Hooks, E., “A Green Vehicle Routing Problem,” *Transportation Research Part E*, Vol. 48, No. 1, 2012, pp. 100–114.
- [6] Schneider, M., Stenger, A., and Goeke, D., “The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations,” *Transportation Science*, Vol. 48, No. 4, 2014, pp. 500–520. doi:http://dx.doi.org/10.1287/trsc.2013.0490.
- [7] Fleischmann, B., *The Vehicle Routing Problem with Multiple Use of Vehicles*, Working paper, Fachbereich Wirtschaftswissenschaften, Universität Hamburg, 1990.
- [8] Koc, C., and Karaoglan, I., “A Branch and Cut Algorithm for the Vehicle Routing Problem with Multiple Use of Vehicles,” *Proceedings of the 41st international conference on computers & industrial engineering*, Los Angeles, USA, 2011, pp. 554–559.
- [9] Clarke, G., and Wright, J. R., “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,” *Operational Research*, Vol. 12, No. 4, 1964, pp. 568–581.
- [10] Paessens, H., “The Savings Algorithm for the Vehicle Rotuing Problem,” *European Journal of Operational Research*, Vol. 34, No. 3, 1988, pp. 336–344.
- [11] Mirshekarian, S., and Süer, G. A., “A Heuristic for the Vehicle Routing Problem with Tight Time Windows and Limited Working Times,” *Proceedings of 26th Annual Production and Operations Management Society Conference*, 2015.
- [12] Laporte, G., Gendreau, M., Potvin, J., and Semet, F., “Classical and Modern Heuristics for the Vehicle Routing Problem,” *International Transactions in Operational Research*, Vol. 7, No. 4–5, 2000, pp. 285–300.